Кодирование типов данных
1С:Предприятие 8 в protobuf

0. Структура пакетов.

Для каждой информационной базы 1С должен создаваться соответствующий набор пакетов protobuf. Это необходимо для корректного отражения и кодирования системы типов информационной базы 1С в protobuf.

В свою очередь набор типов данных, участвующих в интеграции, определяется составом плана обмена 1С. Планов обмена в информационной базе 1С может быть несколько. Другими словами для каждого плана обмена необходимо формировать свой набор пакетов protobuf.

Кроме этого следует учитывать, что разные типы объектов 1С могут иметь одни и те же наименования. Например, справочник и регистр сведений могут иметь одно и то же наименование в рамках одной и той же информационной базы 1С.

Более того, табличные части различных объектов 1С так же могут иметь одинаковые наименования (вложенные типы).

Шаблон идентификации отдельно взятого типа данных 1С:

<Конфигурация>.<ПланОбмена>.**v**<ВерсияСхемы>.<ВидОбъекта>.<ИмяТипа>

Пример: ERP.ПланОбменаKafka.v1.Справочник.Номенклатура

В целях избежания дублирования наименований табличных частей целесообразно использовать вложенные типы данных protobuf. Например, для табличной части "Товары" документа "ЗаказКлиента" наименование типа protobuf будет выглядеть следующим образом:

ERP.ПланОбменаKafka.v1.Документ.ЗаказКлиента.Товары

Учитывая тот факт, что объекты 1С могут ссылаться друг на друга, пакеты protobuf информационной базы должны так же ссылаться друг на друга, используя **import** соответствующих файлов proto.

Структура каталогов файлов для описания схемы protobuf (пример):

```
- ERP
---- ПланОбменаKafka
------- v1
---------- Документ.proto
---------- Справочник.proto
---------- РегистрСведений.proto
---------- РегистрНакопления.proto
```

1. Простые типы данных.

1С	proto3
Неопределенно	google.protobuf.NullValue
Булево	bool
Число	double
Строка	string
Дата	google.protobuf.Timestamp
УникальныйИдентификатор (uuid)	string
Бинарные данные	bytes
`syntax = "proto3";` `package 1c.protobuf.common;`	

ВидДвижения {Приход,Расход}
 	
Предопределённое платформой 1С перечисление, используемое регистрами накопления остатков.
	`enum RecordType`
	`{`
	`  Receipt = 0;`
	`  Expense = 1;`
	`}`

Ссылка, структура **{**int:uuid**}**
 	
Ссылка может быть "пустой", то есть иметь значение нулевого UUID: "00000000-0000-0000-0000-000000000000"
 	
Свойство **type** содержит полное наименование типа данных 1С, например, "Справочник.Номенклатура".
	`message Entity`
	`{`
	`  string type = 1;`
	`  string uuid = 2;`
	`}`
	
	Свойство **uuid** содержит значение внутреннего идентификатора объекта в базе данных 1С.

Составной тип
 	
Составной тип данных 1С полностью соответствует семантике **oneof**.
 	
Может не иметь установленных значений, то есть иметь значение Неопределенно в 1С.
	`message Union`
	`{`
	`  oneof value {`
	`  bool  boolean  = 1;`
	`  double  numeric  = 2;`
	`  string  string  = 3;`
	`  Entity  entity  = 4;`
	`  Timestamp datetime = 5;`
	`  }`
	`}`

УдалениеОбъекта
 	
Информация об удалении объекта из информационной базы 1С.
	`message ObjectDeletion`
	`{`
	`  Entity entity = 1;`
	`}`

Примечание.
Перечисления 1С являются ссылочными типами данных. Их значения кодируются так же, как и тип данных **Ссылка**, то есть как **Entity**. Однако, следует отметить, что значения перечислений это константные значения UUID, определяемые в информационной базе 1С. Одновременно с этим данные значения имеют соответствующие строковые идентификаторы.

2. Сложные типы данных.

2.1. Ссылочные типы данных.

Документы и справочники 1С кодируются в protobuf обычным способом как **message**. При этом для описания их свойств могут быть использованы только все выше описанные простые типы данных.

```
syntax = "proto3";

package ERP.ПланОбменаKafka.v1.Справочник;

import "1c/protobuf/common.proto";

message ДоговорыКонтрагентов
{
   string Ссылка       = 1;
   string Код          = 2;
   string Наименование = 3;
   1c.protobuf.Entity Владелец = 4;
}
```

2.2. Табличные части (вложенные типы).

Табличные части могут иметь только ссылочные типы данных 1С. Они являются неотъемлемой частью своих объектов, образуя таким образом агрегаты. Кодируются как вложенные типы protobuf.

```
message ДоговорыКонтрагентов
{
   message КонтактнаяИнформация
   {
      string НомерТелефона    = 1;
      string ЭлектроннаяПочта = 2;
   }
   repeated КонтактнаяИнформация Контакты = 1;
}
```

2.3. Наборы записей регистров.

Регистры сведений и накопления в 1С всегда изменяются при помощи наборов записей.

Набор записей кодируется как объект, имеющий два свойства: **delete** (отбор) и **insert** (записи набора).

Свойство **delete** содержит ключ записи регистра, по которому выполняется операция удаления.

Свойство **insert** содержит набор записей регистра, которые необходимо вставить в таблицу после выполнения удаления по ключу.

Свойство **insert** может быть пустым, то есть не иметь записей.

Пример описания схемы для регистра сведений "КурсыВалют":

```
syntax = "proto3";

package ERP.ПланОбменаKafka.v1.РегистрСведений;

import "1c/protobuf/common.proto";
import "google/protobuf/timestamp.proto";

message КурсыВалют
{
   message Ключ
   {
      google.protobuf.Timestamp Период = 1;
      1c.protobuf.Entity        Валюта = 2;
   }
   message Запись
   {
      КурсыВалют.Ключ Ключ = 1;
      double          Курс = 2;
   }
           КурсыВалют.Ключ   delete = 1;
   repeated КурсыВалют.Запись insert = 2;
}
```

4. Обязательные заголовки сообщений.

Один топик Kafka может хранить сообщения разных типов для одного и того же объекта метаданных 1С. Например, для справочника "Номенклатура" может существовать запись, как для обычного объекта, так и для типа "УдалениеОбъекта".

Таким образом, целесообразно дополнять сообщения Kafka специальным заголовком, например, **message-type** с указанием его типа.

Кроме этого, необходимо дополнять сообщения значением идентификатора (версии) соответствующей схемы protobuf, например, **schema-version**.